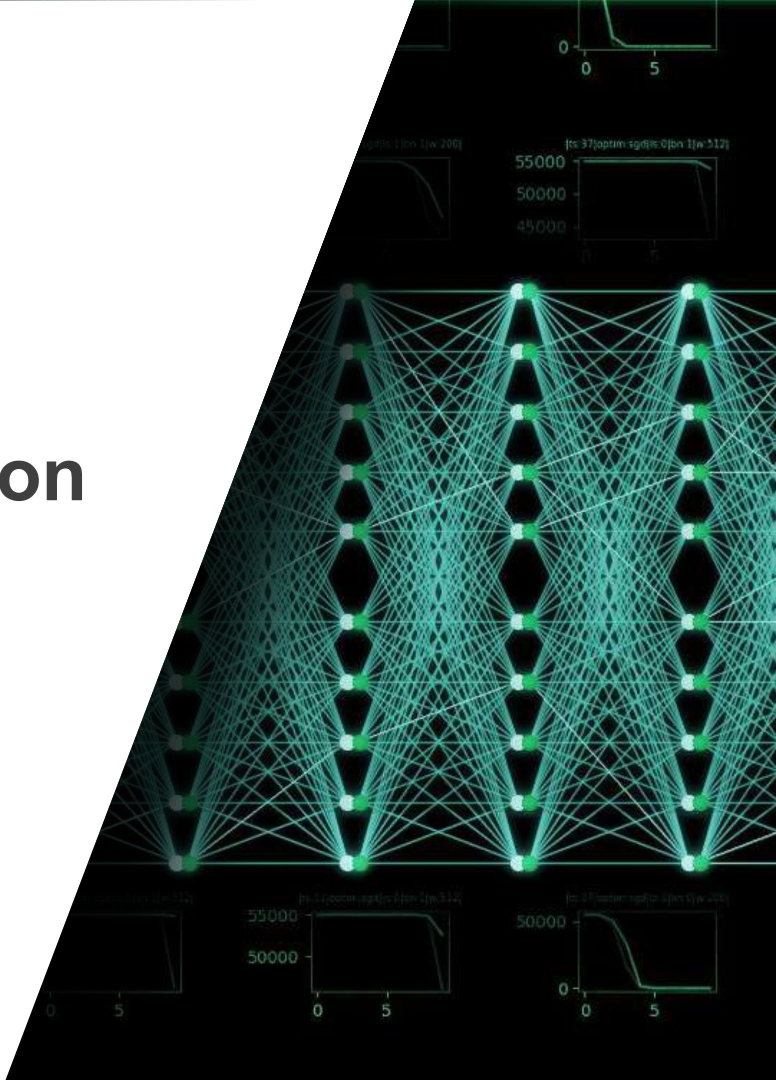
Exploring layerwise decision making in DNNs

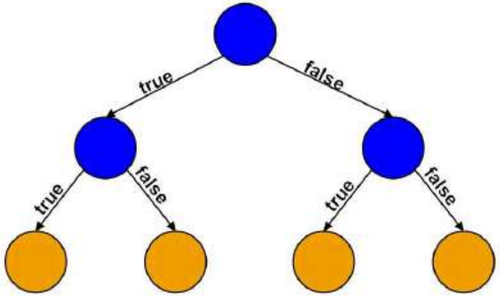
SACAIR2021
Coenraad Mouton and Marelie Davel



[1]Faculty of Engineering, North-West University
[2]Centre for Artificial Intelligence Research (CAIR)

Layerwise decision trees

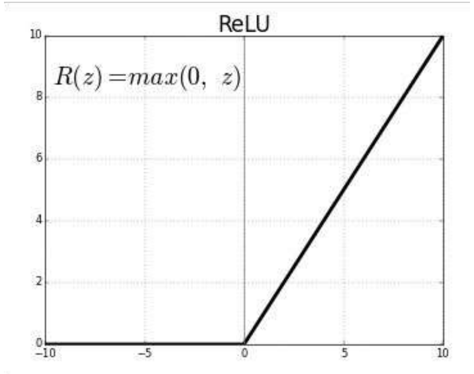
- While very powerful, NNs are still seen as a “black box”
- We developed a method for converting each layer of a neural network to a decision tree
- Decision trees are easily interpretable if they are concise
- We use *binary encodings*



Binary Encodings

- ReLU nodes are either ON or OFF
- Each node partitions the samples of the training set
- ON for certain samples, OFF for others

- *Binary encoding for a layer*: A string of binary values, a 1 or 0 for each node for a specific sample.
- Can be encoded as a vector
- E.g. [1 0 1 1 1 0]



DNNs as layers of cooperating classifiers[1]

- Show that in shallower layers, almost each sample has a unique binary encoding
- In deeper layers, all encodings become class specific
- Samples of the same class share binary encodings

1. Davel et. al. Proceedings of the AAAI Conference on Artificial Intelligence (Apr 2020).

What we did

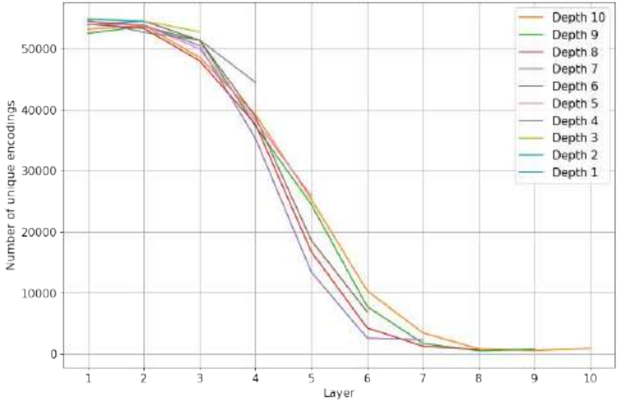
- Confirmed the results of “DNNs as layers of cooperating classifiers”
- Use these encodings to build layerwise decision trees
- Measured the accuracy and size of decision tree at each layer
- Visualized these decision trees with a mean-sample heatmap method

Networks

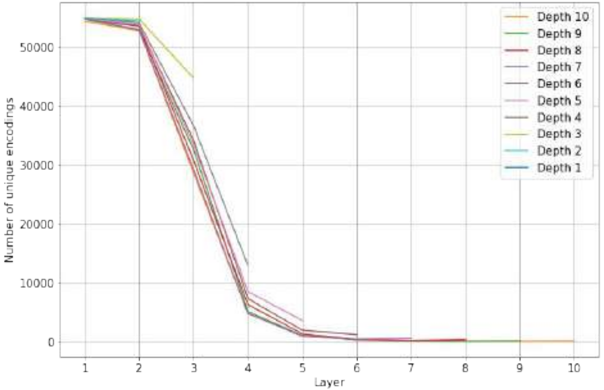
- MNIST and FMNIST data sets
- 4 pairs of 10 MLPs - 1 to 10 hidden layers in depth
  - MNIST-100: 1 to 10 hidden layers with a width of 100
  - MNIST-20: 1 to 10 hidden layers with a width of 20
  - FMNIST-100: 1 to 10 hidden layers with a width of 100
  - FMNIST-40: 1 to 10 hidden layers with a width of 40
- Then measure the number of unique binary encodings, per layer, for all of these networks
- All networks are trained to interpolation: 100% train accuracy
- Each data set consists of 55,000 training samples

Unique Encodings per layer

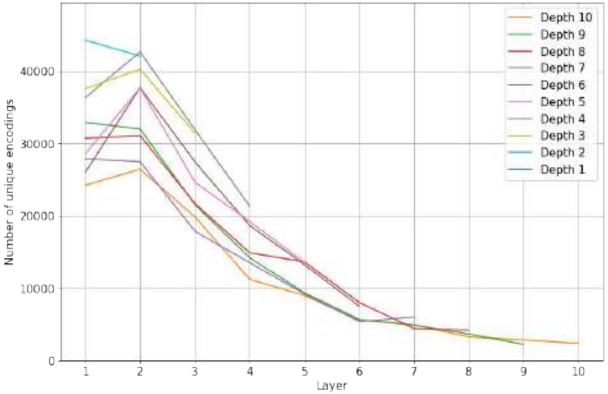
FMNIST-100



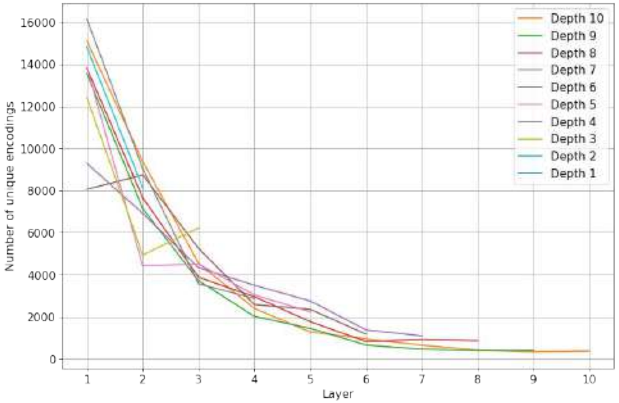
MNIST-100



FMNIST-40



MNIST-20



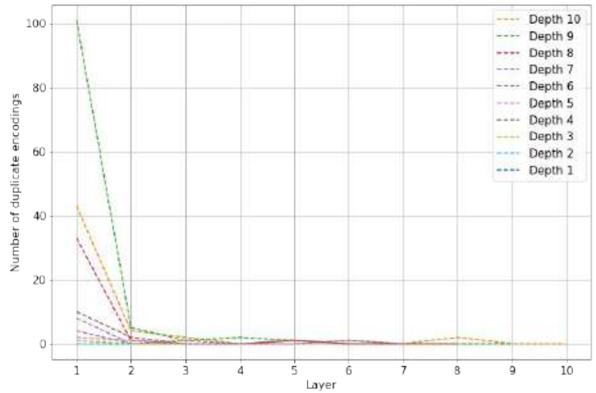
Duplicates

A *duplicate* encoding: An encoding that is shared by samples of more than a single class.

Need to take note of them if we wish to derive rules/decision trees from binary encodings.

Duplicate Encodings per layer

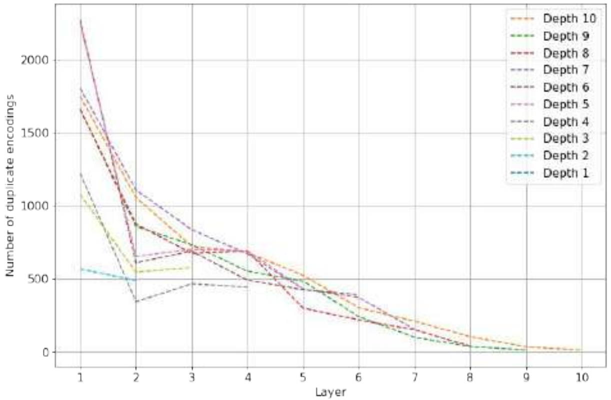
FMNIST-100



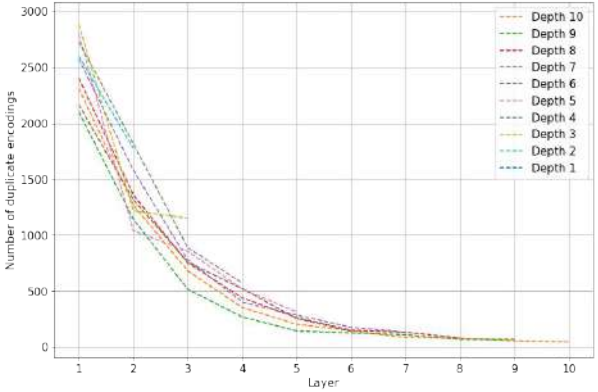
MNIST-100

0 duplicates

FMNIST-40



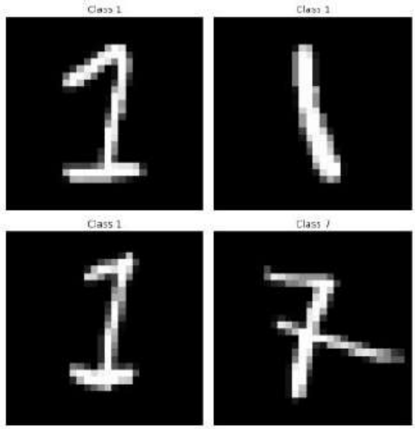
MNIST-20



NWU

What causes duplicates?

- *Idiosyncratic samples*
- Samples which don't look like their class majority
- Share an encoding with more typical looking samples
- In part, due to visual overlap



45 samples of class 1 share an encoding with 1 idiosyncratic 7

Decision tree induction - Algorithm

Use binary encodings to build a decision tree:

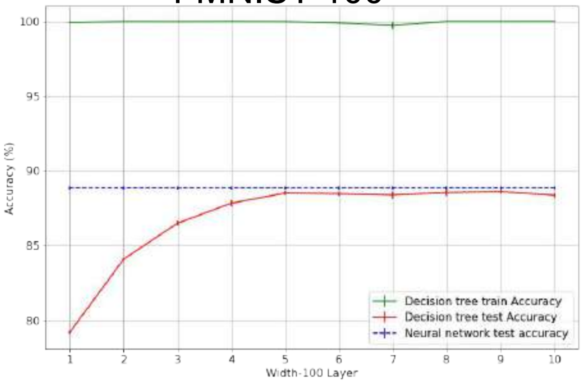
1. Find all the unique and duplicate encodings at a layer for all training set samples
2. Label each encoding according to its class, if there is more than a single class (a duplicate): label it according to the class-majority.
3. Apply a thresholding operation - remove each encoding that doesn't occur for at least *threshold* number of samples. Threshold becomes an important hyperparameter
4. Build a decision tree using this data set

Decision tree results

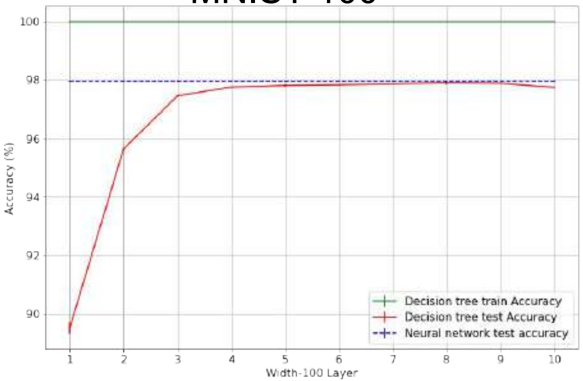
- Use the 10-layer MNIST and FMNIST architectures
- Following set of results is with a threshold of 0 (no encodings are disregarded)

Decision tree results

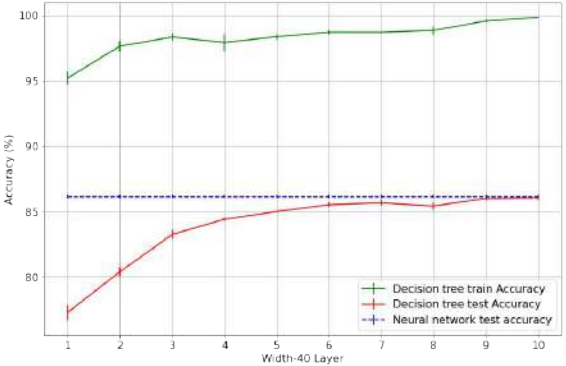
FMNIST-100



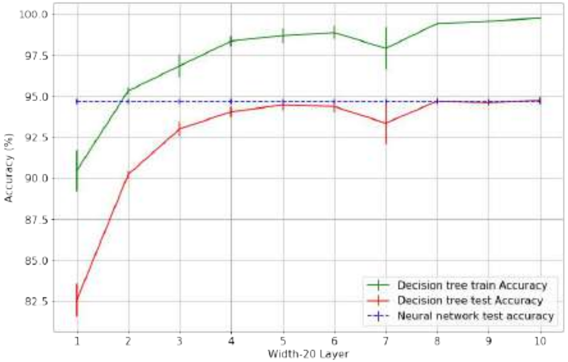
MNIST-100



FMNIST-40



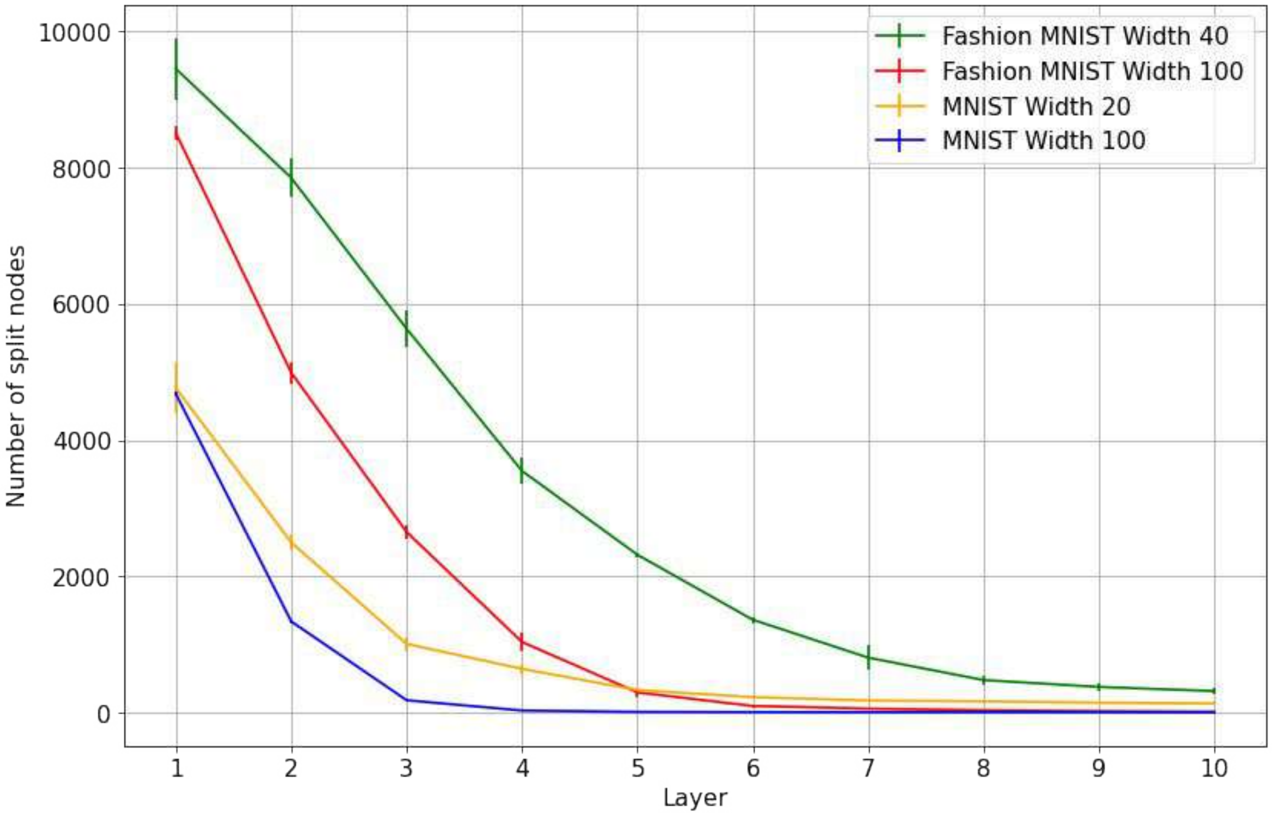
MNIST-20



Decision tree results

- We also measure the size of each decision tree by finding the number of ‘split nodes’ for each.
- Meaning: Each node in the tree that is not a leaf node is counted as a ‘split node’.

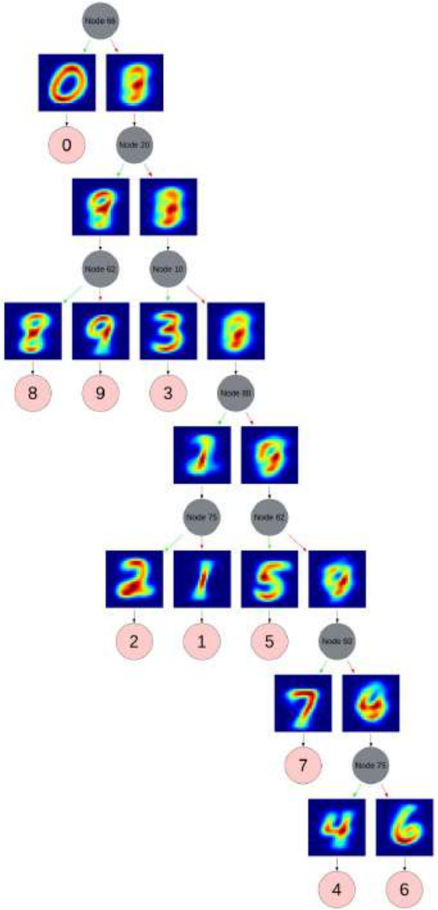
Decision tree results



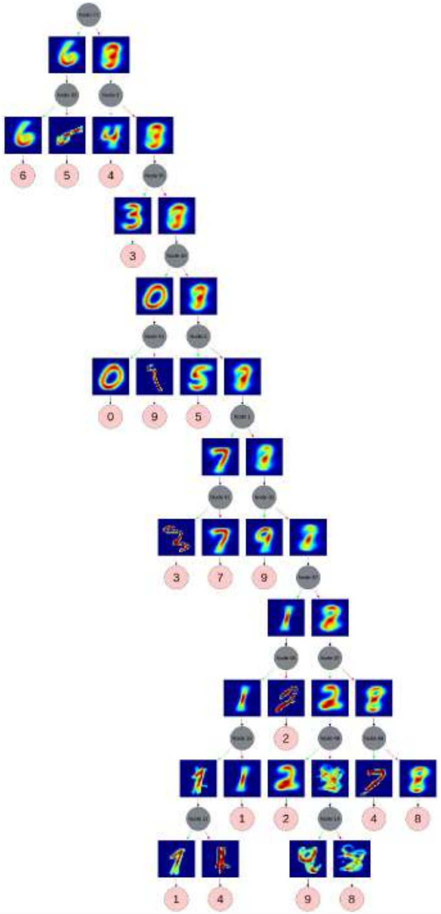
Decision tree: Visualization

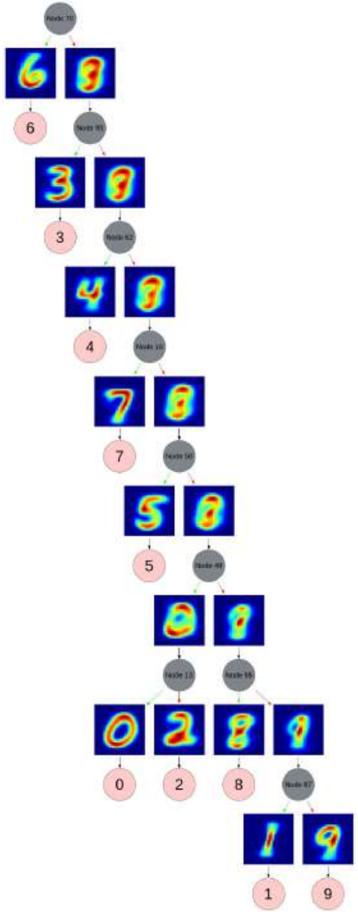
- Each split node shows 2 heatmaps: One for the ON state of the node, and one for the OFF state.
- ON is indicated by a green edge, OFF by a red edge.
- Heatmaps are the average of all the samples for which the unique combination of nodes are ON or OFF
- Leaf nodes are shown as pink circles indicating the classification.

Decision tree: Visualization: MNIST-100 layer 10 no threshold



Decision tree: Visualization: MNIST-100 layer 5 no threshold and threshold 100



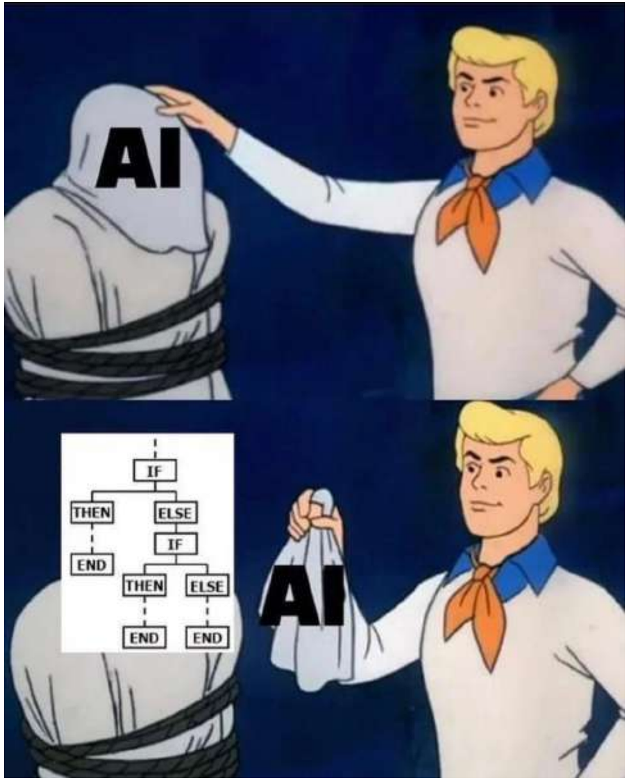


Conclusion

- Binary encodings can be used to derive decision trees, and these decision trees are able to provide excellent accuracy on both the train and test data sets.
- The decision trees can be used to visualize the decision-making process of a model. Demonstrated here with mean sample heatmaps, the heatmaps can be replaced with those produced by a variety of existing feature attribution methods.

Questions?

Thank you



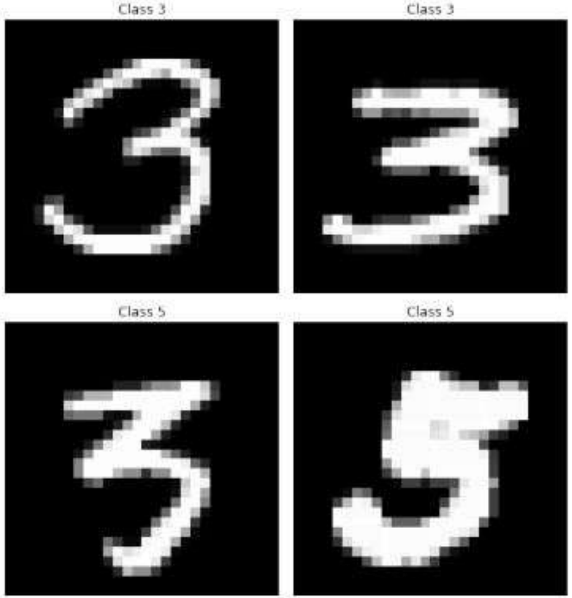
Training hyperparameters

- Adam
- Cross Entropy
- Initial learning rate chosen empirically
- Learning rate decay chosen empirically
- To interpolation (100% train accuracy)
- No early stop
- No batch norm
- Bias on first layer only

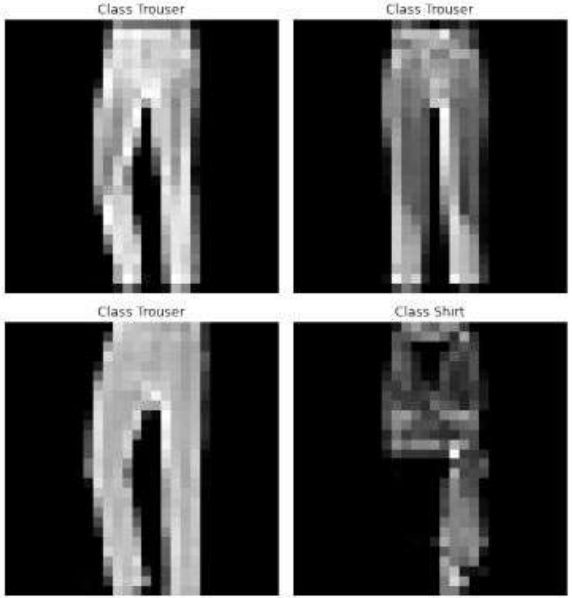
Data set	Width	Learning rate	Decay gamma	Decay step size	Train accuracy (%)	Test accuracy (%)
MNIST	100	0.001	0.990	5	100.00	97.82 to 98.19
FMNIST	100	0.001	0.990	5	100.00	87.96 to 89.10
MNIST	20	0.002	0.500	100	100.00	94.62 to 96.12
FMNIST	40	0.002	0.850	50	100.00	85.23 to 86.70

Duplicates: Idiosyncratic to typical

Case A - Idiosyncratic-to-many



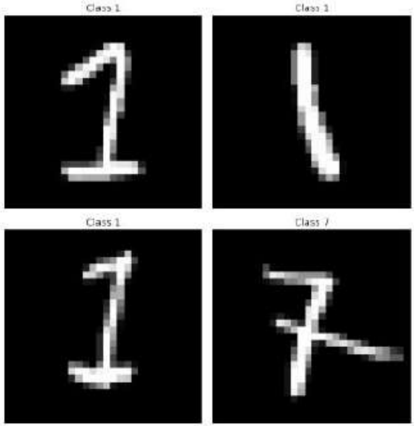
4 idiosyncratic (class 5) with many typical samples 3221 (class 3)



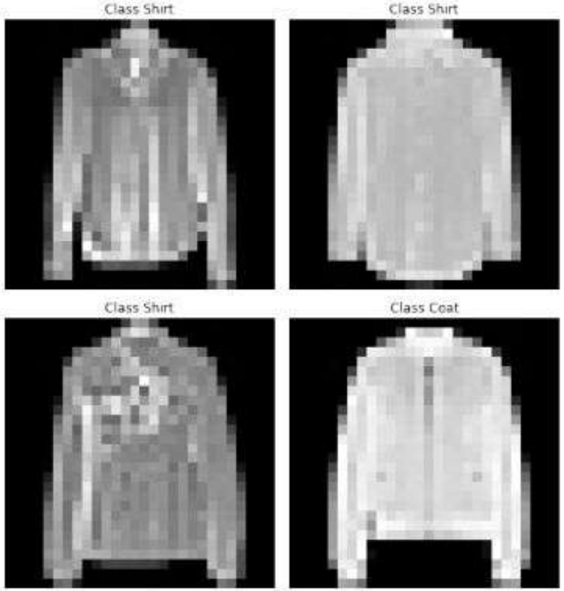
4 idiosyncratic (class 5) with many typical samples 3221 (class 3)

Duplicates: Idiosyncratic to typical

Case B - Idiosyncratic-to-few



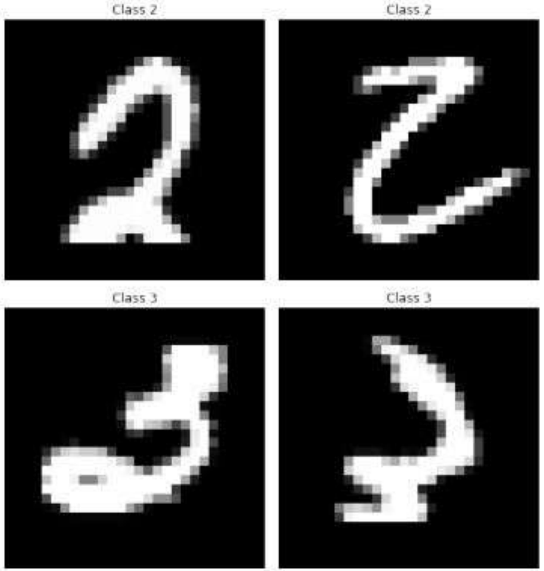
45 samples of class 1 share an encoding with 1 idiosyncratic 7



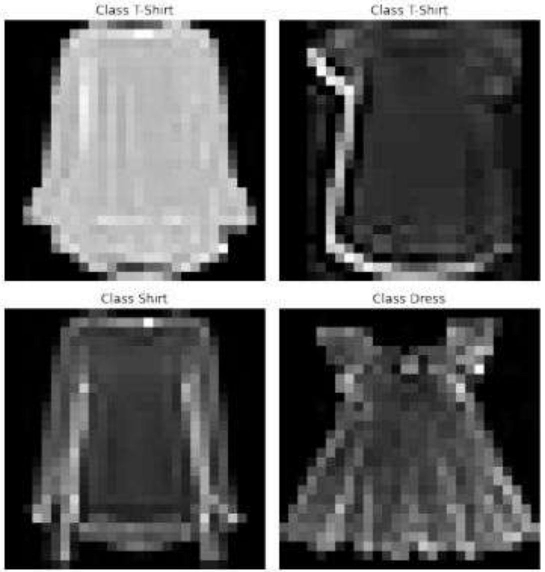
53 typical example of ‘shirt’ share an encoding with 1 idiosyncratic sample of “coat

NWU

Duplicates: Idiosyncratic to Idiosyncratic



Four idiosyncratic samples: 2 of class 2 and 3



4 idiosyncratic samples: 2 of ‘T-shirt’, 1 of ‘Shirt’ and 1 of ‘dress’

Decision tree details

- Use the CART (Classification and Regression Trees) implementation of sci-kit learn
- Don't apply any additional pruning or limitations to the tree
- Use the gini index for calculating the splitting criterion (can also use entropy, doesn't make a difference)
- We treat the data set as "balanced" so if there are more/less encodings for a specific class, this won't influence the split-point calculation of the decision tree

Decision tree: Thresholding

- How does thresholding effect accuracy and size?
- Use the MNIST-100 network as an example
- Show the number of split nodes, and also number of encodings in the data set after applying threshold of 0, 100, and 1000.
- Meaning we disregard any encoding that does not occur for at least 0, 100, or 1000 samples respectively.

Decision tree: Thresholding

Total	Layer	Threshold		
		0 samples	100 samples	1000 samples
Encodings	1	54,739	0	0
Split nodes	1	4,682	0	0
Encodings	2	52,265	0	0
Split nodes	2	1,341	0	0
Encodings	3	26,357	27	0
Split nodes	3	184	6	0
Encodings	4	5,236	80	10
Split nodes	4	33	9	7
Encodings	5	900	62	14
Split nodes	5	14	9	9
Encodings	6	342	39	17
Split nodes	6	10	9	9
Encodings	7	161	32	13
Split nodes	7	9	9	9
Encodings	8	138	25	17
Split nodes	8	9	9	9
Encodings	9	187	34	13
Split nodes	9	9	9	9
Encodings	10	196	32	15
Split nodes	10	9	9	9

Data Set	Layer	Accuracy (%)		
		Threshold 0	Threshold 100	Threshold 1,000
Train	1	100.00	0.00	0.00
Test	1	89.60	0.00	0.00
Train	2	100.00	0.00	0.00
Test	2	95.67	0.00	0.00
Train	3	100.00	61.50	0.00
Test	3	97.47	61.05	0.00
Train	4	100.00	97.08	77.19
Test	4	97.77	94.79	75.58
Train	5	100.00	98.99	94.33
Test	5	97.85	96.41	92.08
Train	6	100.00	99.33	99.06
Test	6	97.84	96.88	96.65
Train	7	100.00	99.66	98.32
Test	7	97.85	97.29	95.67
Train	8	100.00	99.76	98.55
Test	8	97.85	97.41	96.03
Train	9	100.00	99.84	99.19
Test	9	97.78	97.48	96.87
Train	10	100.00	99.75	97.97
Test	10	97.81	97.28	95.75

Decision tree: Thresholding

Total	Layer	Threshold		
		0 samples	100 samples	1000 samples
Encodings	1	54,739	0	0
Split nodes	1	4,682	0	0
Encodings	2	52,265	0	0
Split nodes	2	1,341	0	0
Encodings	3	26,357	27	0
Split nodes	3	184	6	0
Encodings	4	5,236	80	10
Split nodes	4	33	9	7
Encodings	5	900	62	14
Split nodes	5	14	9	9
Encodings	6	342	39	17
Split nodes	6	10	9	9
Encodings	7	161	32	13
Split nodes	7	9	9	9
Encodings	8	138	25	17
Split nodes	8	9	9	9
Encodings	9	187	34	13
Split nodes	9	9	9	9
Encodings	10	196	32	15
Split nodes	10	9	9	9

Data Set	Layer	Accuracy (%)		
		Threshold 0	Threshold 100	Threshold 1,000
Train	1	100.00	0.00	0.00
Test	1	89.60	0.00	0.00
Train	2	100.00	0.00	0.00
Test	2	95.67	0.00	0.00
Train	3	100.00	61.50	0.00
Test	3	97.47	61.05	0.00
Train	4	100.00	97.08	77.19
Test	4	97.77	94.79	75.58
Train	5	100.00	98.99	94.33
Test	5	97.85	96.41	92.08
Train	6	100.00	99.33	99.06
Test	6	97.84	96.88	96.65
Train	7	100.00	99.66	98.32
Test	7	97.85	97.29	95.67
Train	8	100.00	99.76	98.55
Test	8	97.85	97.41	96.03
Train	9	100.00	99.84	99.19
Test	9	97.78	97.48	96.87
Train	10	100.00	99.75	97.97
Test	10	97.81	97.28	95.75

Decision tree: Thresholding

Total	Layer	Threshold		
		0 samples	100 samples	1000 samples
Encodings	1	54,739	0	0
Split nodes	1	4,682	0	0
Encodings	2	52,265	0	0
Split nodes	2	1,341	0	0
Encodings	3	26,357	27	0
Split nodes	3	184	6	0
Encodings	4	5,236	80	10
Split nodes	4	33	9	7
Encodings	5	900	62	14
Split nodes	5	14	9	9
Encodings	6	342	39	17
Split nodes	6	10	9	9
Encodings	7	161	32	13
Split nodes	7	9	9	9
Encodings	8	138	25	17
Split nodes	8	9	9	9
Encodings	9	187	34	13
Split nodes	9	9	9	9
Encodings	10	196	32	15
Split nodes	10	9	9	9

Data Set	Layer	Accuracy (%)		
		Threshold 0	Threshold 100	Threshold 1,000
Train	1	100.00	0.00	0.00
Test	1	89.60	0.00	0.00
Train	2	100.00	0.00	0.00
Test	2	95.67	0.00	0.00
Train	3	100.00	61.50	0.00
Test	3	97.47	61.05	0.00
Train	4	100.00	97.08	77.19
Test	4	97.77	94.79	75.58
Train	5	100.00	98.99	94.33
Test	5	97.85	96.41	92.08
Train	6	100.00	99.33	99.06
Test	6	97.84	96.88	96.65
Train	7	100.00	99.66	98.32
Test	7	97.85	97.29	95.67
Train	8	100.00	99.76	98.55
Test	8	97.85	97.41	96.03
Train	9	100.00	99.84	99.19
Test	9	97.78	97.48	96.87
Train	10	100.00	99.75	97.97
Test	10	97.81	97.28	95.75

Decision tree: Thresholding

Total	Layer	Threshold		
		0 samples	100 samples	1000 samples
Encodings	1	54,739	0	0
Split nodes	1	4,682	0	0
Encodings	2	52,265	0	0
Split nodes	2	1,341	0	0
Encodings	3	26,357	27	0
Split nodes	3	184	6	0
Encodings	4	5,236	80	10
Split nodes	4	33	9	7
Encodings	5	900	62	14
Split nodes	5	14	9	9
Encodings	6	342	39	17
Split nodes	6	10	9	9
Encodings	7	161	32	13
Split nodes	7	9	9	9
Encodings	8	138	25	17
Split nodes	8	9	9	9
Encodings	9	187	34	13
Split nodes	9	9	9	9
Encodings	10	196	32	15
Split nodes	10	9	9	9

Data Set	Layer	Accuracy (%)		
		Threshold 0	Threshold 100	Threshold 1,000
Train	1	100.00	0.00	0.00
Test	1	89.60	0.00	0.00
Train	2	100.00	0.00	0.00
Test	2	95.67	0.00	0.00
Train	3	100.00	61.50	0.00
Test	3	97.47	61.05	0.00
Train	4	100.00	97.08	77.19
Test	4	97.77	94.79	75.58
Train	5	100.00	98.99	94.33
Test	5	97.85	96.41	92.08
Train	6	100.00	99.33	99.06
Test	6	97.84	96.88	96.65
Train	7	100.00	99.66	98.32
Test	7	97.85	97.29	95.67
Train	8	100.00	99.76	98.55
Test	8	97.85	97.41	96.03
Train	9	100.00	99.84	99.19
Test	9	97.78	97.48	96.87
Train	10	100.00	99.75	97.97
Test	10	97.81	97.28	95.75